REMARKS

Reconsideration of this application, based on this amendment and these following remarks, is respectfully requested.

Claims 2 through 17 remain in this case. Amendment to claims 2, 6, 7, 12, 13, and 15 is presented.

Claims 2 through 4, 6, and 12 through 15 were finally rejected under §103 as unpatentable over the Kiuchi et al. reference[1] in view of the Osovets reference[2]. Claims 5, 7, 8, 16, and 17 were finally rejected over the Kiuchi et al. and Osovets reference as applied against the above claims, and further in view of the George reference[3]. Claims 9 through 11 were finally rejected under §103 as unpatentable over the Kiuchi et al. reference in view of the Hennessy reference[4].

Amendment to claim 2 is presented to clarify its patentability over the applied references. Specifically, claim 2 is amended to now recite that the loop cache control logic is also for controlling the branch cache register file to load an instruction code received at its data input from the program memory responsive to receiving a backward branch signal in combination with the fetch address not corresponding to one of the instruction codes stored in the branch cache register file. The proposed amendment recites the verb "load" rather than "store", to clarify the distinction between the claimed processor and the prior art. The specification clearly supports this proposed amendment,[5] and as such no new matter is presented by this amendment to claim 2.

[1] U.S. Patent No. 5,579,493, issued November 26, 1996 to Kiuchi et al..

[2] U.S. Patent No. 6,125,440, filed May 21, 1998, and issued September 26, 2000 to Osovets.

[3] U.S. Patent No. 4,626,988, issued December 2, 1986 to George.

[4] Hennessy et al., *Computer Organization and Design – The Hardware/Software Interface* (2d. ed., Morgan Kaufmann Publishers, Inc., 1998), pp. 542, 545, 549, 579.

[5] Specification of S.N. 09/713,731, page 20, line 20 through page 21, line 9; page 28, lines 12 through 23; page 30, lines 8 through 25.

Claims 6 and 7 are proposed to be similarly amended, for consistency with the proposed amendment to claim 2, upon which they depend.

Proposed amended claim 2 is directed to an instruction-programmable processor having, *inter alia*, that the loop cache control logic is for controlling the branch cache register file to load an instruction code received at its data input from the program memory responsive to receiving a backward branch signal in combination with the fetch address not corresponding to one of the instruction codes stored in the branch cache register file. According to the invention of claim 2, therefore, the loop cache control logic enables the loading of instructions in the branch cache register file in response to backward branches, for example as a "loop front cache"[6] or as a "loop tail cache"[7] for which the instruction codes are not already stored in the loop cache (*i.e.*, the "miss" condition of the fetch address not matching an instruction code already stored in the branch cache register file). Because of this construction and operation, the processor of claim 2 and its dependent claims advantageously minimizes the number of accesses to memory, in an automatic manner even for nested program loops, and minimizes the loading, into its cache loop, instructions that will only be executed once.[8]

In the previous Amendment,[9] Applicant urged that the combined teachings of the references fall short of the requirements of claim 2 and its dependent claims, because none of the references disclose loop cache control logic with the recited functionality regarding the control of the branch cache register file to store instruction codes in the event of backward branches and non-matching fetch address. But in the Office Action of September 21, the Examiner found this argument to not be persuasive.[10] Specifically, the Examiner again asserted that the Osovets reference teaches[11] that the repeated or looped instructions are "stored or

[6] Specification, *supra*, page 14, line 20 *et seq.*; page 20, line 20 through page 21, line 28.
[7] Specification, *supra*, page 22, line 15 *et seq.*; page 32, line 3 through page 33, line 2.
[8] Specification, *supra*, page 5, lines 2 through 12; page 14, lines 20 through 26; page 21, lines 22 through 28; page 32, line 13 through page 33, line 8.
[9] Amendment of February 9, 2004.
[10] Office Action of September 21, 2004, page 25, ¶27.
[11] Osovets, *supra*, column 9, lines 34 through 51.

retained" in the shift register (corresponding to the branch cache register file) in response to a jump back or branch backwards instruction.[12]

It is instructive to again review the detailed basis for the §103 rejection of claim 2 and its dependent claims. The Examiner asserts that the Kiuchi et al. reference teaches the storing of an instruction code, in a branch cache register file, responsive to receiving a signal in combination with a fetch address not corresponding to an instruction code stored in the branch cache register file,[13] and that this signal is generated by a repeat instruction generating a repeat signal, "and not a backward branch generating a backward branch signal".[14] The Examiner then asserted that the Osovets reference teaches that repeated or looped instructions "are stored or retained" in a shift register (corresponding to the claimed branch cache register file) upon decoding of a jump back or branch backward instruction.[15] Motivation to modify the Kiuchi et al. teachings with these teachings of the Osovets reference was found by the Examiner,[16] and the claims were rejected accordingly.

Proposed amended claim 2 clarifies its distinction over this combination of the prior art by now reciting that the loop cache control logic is also for controlling the branch cache register file to *load* an instruction code received at its data input from the program memory responsive to receiving a backward branch signal in combination with the fetch address not corresponding to one of the instruction codes stored in the branch cache register file. Applicant respectfully submits that, by now reciting that the branch cache register file loads an instruction code responsive to this combination of events, the distinction between proposed amended claim 2 and the basis of the rejection by the Examiner is clarified. Specifically, while the claim words "to store" could be interpreted to include "retaining",[17] which is in fact what is taught by the reference, the action verb "to load" in proposed amended claim 2 can in no way be met by the disclosed action of "retaining".

[12] Office Action, *supra*, page 25, ¶27.
[13] Office Action, *supra*, page 5, ¶8.a.v.
[14] Office Action, *supra*, page 5, ¶8.b.
[15] Office Action, *supra*, pp. 5-6, ¶8.c; page 25, ¶27.
[16] Office Action, *supra*, page 6, ¶8.d.

Applicant respectfully submits that, upon entry of this amendment, amended claim 2 and its dependent claims will be patentably distinct over the applied references, on the grounds that the combined teachings of the applied references fall short of the requirements of the claims.

Applicant agrees with the Examiner that the Kiuchi et al. reference fails to disclose the loading and storing of instruction codes in a branch cache register file. However, Applicant now respectfully submits that the Osovets reference fails to disclose the loading of the branch cache register file with an instruction code responsive to receiving a backward branch signal in combination with a fetch address not corresponding to one of the instruction codes stored in the branch cache register file, as now required by proposed amended claim 2.

Rather, Applicant respectfully submits that the Osovets reference teaches the loading of every instruction code into its shift register, regardless of whether a backward branch has been taken, and the retaining (without loading new instruction codes) of those instruction codes upon detecting such a backward branch. The Osovets reference describes its signal line SH_EN as enabling the shift register to store to in its registers, or shift in, the information at its input.[18] The Osovets reference further discloses that, upon reset and initialization, this shift register enable signal SH_EN is enabled so that instructions are fetched into the shift register bank 24 when fetched into the instruction register.[19] And the reference goes on to teach that, responsive to a "jump back" instruction, the shift register enable signal SH_EN is then *disabled* so that shift register bank 24 retains those instruction codes that are currently stored therein.[20] In other words, the Osovets reference teaches the unconditional loading of instruction codes into its shift register 24 *until* a backward branch is detected, at which time the loading stops and the then-current contents of the shift register are retained.

[17] Applicant does not necessarily agree with such an interpretation, but instead presents this amendment to avoid the issue of claim interpretation in this regard.
[18] Osovets, *supra,* column 4, lines 46 through 48.
[19] Osovets, *supra,* column 8, lines 60 through 64.
[20] Osovets, *supra,* column 9, lines 34 through 44.

In contrast, proposed amended claim 2 requires the loop cache control logic to control the branch cache register file to load an instruction code responsive to a backward branch signal in combination with the fetch address not corresponding to one of the stored instruction codes. In effect, this recited operation is the opposite of that taught by the Osovets reference, in which the loading continues until a backward branch instruction, at which point the loading stops.

Accordingly, neither of the Kiuchi et al. and Osovets references teach the controlling of the branch cache register file by the loop cache control logic as required by proposed amended claim 2. And none of the other references of record disclose this function. Accordingly, Applicant respectfully submits that, upon entry of this amendment, the combined teachings of the prior art of record in this case fall short of the requirements of claims 2 through 11.

As previously argued,[21] Applicant further respectfully submits that there is no suggestion to modify these teachings in such a manner as to reach claim 2 and its dependent claims. The Kiuchi et al. reference requires the programmer to use a specific "repeat" instruction for its operation. While the Osovets reference does not require such an instruction, each fetched instruction is simply unconditionally loaded into also the shift register bank, and the backward branch then enables *access* of these stored instructions from the shift register bank. There is no suggestion from these references, nor from the other prior art, to load the instructions in the branch cache register file responsive to a backward branch, as required by proposed amended claim 2 and its dependent claims.

The advantages provided by the processor of proposed amended claim 2, including the ability to closely cache the instructions for nested program loops, and the minimizing of the caching of instructions that are executed only once, arise directly from the difference between the claim and the prior art, negating suggestion from the prior art to modify its teachings in such a manner as to reach the claim, and indicating the value of the inventive processor.

For these reasons, Applicant respectfully submits that claim 2 and its dependent claims are patentably distinct over the applied references, taken individually or in any proper

[21] Amendment, *supra*.

combination. Entry of the proposed amendment to claim 2, and favorable reconsideration of it and its dependent claims, are therefore respectfully requested.

As mentioned above, claim 6 is also proposed to be amended to now further recite, relative to proposed amended claim 2 upon which it depends, that the base address register is for loading the contents of the next candidate register as a base fetch address responsive to the loop cache control logic receiving a backward branch signal in combination with the fetch address corresponding to the contents of the next candidate register, and that the loop cache control logic is for controlling the branch cache register file to load an instruction code received at its data input from the program memory responsive to receiving a backward branch signal in combination with the fetch address corresponding to the contents of the next candidate register. This proposed amendment to claim 6 is clearly supported by the specification,[22] and therefore no new matter is presented.

As discussed above relative to proposed amended claim 2, upon which claim 6 depends, there is no teaching in the applied prior art of the loading of instruction codes in the branch cache register file in response to a backward branch in combination with the fetch address not corresponding to one of the instruction codes stored in the branch cache register file. In addition, because this responsive loading is not disclosed, there is necessarily no teaching in that prior art of the additional limitations of proposed amended claim 6, especially the next candidate register and the additionally recited functions of the base address register and the loop cache control logic. Furthermore, considering the lack of suggestion of the loading of instruction codes responsive to a backward branch, as discussed above, there is especially no suggestion from the prior art to modify its combined teachings to provide the "loop tail cache" functionality that results from the construction of proposed amended claim 6 and its dependent claims.

[22] Specification, *supra*, page 28, lines 12 through 23; page 30, lines 8 through 25.

For these reasons, and for additional reasons previously argued[23], Applicant respectfully submits that proposed amended claim 6 is also patentably distinct over the prior art of record in this case.

Regarding claims 9 through 11, Applicant respectfully submit that these claims are also patentably distinct over the applied combination of references, for the same reasons as discussed above relative to proposed amended claim 2, upon which these claims depend. More specifically, Applicant submits that there is no teaching in the Hennessy et al. reference, which was applied specifically against claims 9 through 11, regarding the loading of the instruction codes in the branch cache register file in response to a backward branch in combination with the fetch address not corresponding to one of the instruction codes stored in the branch cache register file, as required by proposed amended claim 2 upon which these claims depend.

For these reasons, Applicant respectfully submits that claims 9 through 11 are also patentably distinct over the prior art of record in this case.

Claim 12 is proposed to be amended in a similar manner as discussed above relative to claim 2. Specifically, proposed amended claim 12 is directed to a method of fetching instructions for execution in an instruction-programmable processor, and now requires the steps of fetching the instruction code corresponding to the fetch address from program memory, loading the instruction code in a next indexed location of the branch cache register file, and setting a valid bit corresponding to the next indexed location, all responsive to receiving a fetch address following the backward branch operation and within a storage capacity of the branch cache register file. Similarly as discussed above relative to claim 2, the specification of this application clearly supports this amendment to claim 12,[24] and as such no new matter is presented by this amendment.

Claims 13 and 15 are also proposed to be amended in a similar fashion, for consistency with claim 12 upon which they depend.

[23] *See* Amendment of February 9, *supra*.

[24] Specification, *supra*, page 20, line 20 through page 21, line 9; page 28, lines 12 through 23; page 30, lines 8 through 25.

Similarly as discussed above relative to claim 2, Applicant submits that this proposed amendment to claim 12 clarifies its patentability over the applied references in this case.

As mentioned above, the Examiner admits that the Kiuchi et al. reference fails to disclose the loading of a branch cache register file with an instruction code responsive to receiving a fetch address following the backward branch operation and within a storage capacity of the branch cache register file, as required by proposed amended claim 12.

Applicant further respectfully submits that the Osovets reference also fails to disclose this step. Instead, as mentioned above, the Osovets reference teaches loading every instruction code into its shift register, regardless of whether a backward branch has been taken. Upon the detecting of a backward branch, the Osovets reference teaches the retaining of those instruction codes that have already been loaded into the shift register, and inhibiting the loading of new instruction codes. This operation is disclosed in the Osovets reference relative to its shift register enable signal SH_EN that, upon reset, is enabled so that instructions are fetched into the shift register bank 24 when fetched into the instruction register,[25] and that is disabled in responsive to a "jump back" instruction, such that shift register bank 24 then retains those instruction codes that it currently stores.[26] In summary, the Osovets reference teaches unconditionally loading instruction codes into its shift register *until* a backward branch is taken, after which the loading stops and the then-current contents of the shift register are retained.

Accordingly, neither of the Kiuchi et al. and Osovets references teach the loading of the branch cache register file with instruction codes responsive to receiving a fetch address following a backward branch operation, as required by proposed amended claim 12. And none of the other references of record disclose this function. Accordingly, Applicant respectfully submits that, upon entry of this amendment, the combined teachings of the prior art of record in this case fall short of the requirements of claims 12 through 17.

[25] Osovets, *supra*, column 8, lines 60 through 64.

[26] Osovets, *supra*, column 9, lines 34 through 44.

Applicant further respectfully submits that there is no suggestion to modify these teachings in such a manner as to reach claim 12 and its dependent claims. The Kiuchi et al. reference lacks such suggestion, because it requires the use of a specific "repeat" instruction to enable its functionality, and the Osovets reference lacks such suggestion because of its unconditional loading of instruction codes into the shift register bank, with backward branches then enabling *access* of these stored instructions from the shift register bank. There is no suggestion from these references, nor from the other prior art, to load the instructions in the branch cache register file responsive to any backward branch, as required by proposed amended claim 12 and its dependent claims.

The advantages provided by the processor of proposed amended claim 12, especially in minimizing caching of instructions that are executed only once, are due directly to the difference between the claim and the prior art, and therefore support the patentability of these claims. Accordingly, Applicant respectfully submits that there is no suggestion in the prior art to modify the teachings of these references in such a manner as to reach the claim.

For these reasons, Applicant respectfully submits that proposed amended claim 12 and its dependent claims are patentably distinct over the applied references, taken individually or in any proper combination. Entry of the proposed amendment to claim 12, and favorable reconsideration of it and its dependent claims, are therefore respectfully requested.

As mentioned above, claim 15 is also proposed to be amended to now further recite, relative to proposed amended claim 12 upon which it depends, that the base address register is for loading the contents of the next candidate register as a base fetch address responsive to the loop cache control logic receiving a backward branch signal in combination with the fetch address corresponding to the contents of the next candidate register, and that the loop cache control logic is for controlling the branch cache register file to load an instruction code received at its data input from the program memory responsive to receiving a backward branch signal in combination with the fetch address corresponding to the contents of the next candidate register.

This proposed amendment to claim 6 is clearly supported by the specification,[27] and therefore no new matter is presented.

As discussed above relative to proposed amended claim 12, upon which claim 15 depends, there is no teaching in the applied prior art of the loading of instruction codes in the branch cache register file responsive to receiving a fetch address following the backward branch operation to within a storage capacity of the branch cache register file. And lacking any teaching of this responsive loading is not disclosed, necessarily there is no teaching in the prior art of the additional limitations of proposed amended claim 15, especially the comparing of a fetched address to the contents of a candidate register and the additionally recited functions of the loading the candidate register. Furthermore, considering that the prior art lacks any suggestion to load instruction codes responsive to a backward branch, as discussed above, there is especially no suggestion from the prior art to modify its combined teachings to provide the "loop tail cache" functionality that results from the construction of proposed amended claim 15 and its dependent claims.

The prior art again cited as pertinent but not applied has been considered, but is not felt to come within the scope of the claims in this case.

[27] Specification, *supra*, page 28, lines 12 through 23; page 30, lines 8 through 25.

For the above reasons, Applicant respectfully submits that, upon entry of this amendment, all claims in this case will be in condition for allowance. Reconsideration of the above-referenced application is therefore respectfully requested.

Respectfully submitted,

Rodney M. Anderson
Registry No. 31,939
Attorney for Applicant

Anderson, Levine & Lintel, L.L.P.
14785 Preston Road, Suite 650
Dallas, Texas 75254
(972) 664-9554

CERTIFICATE OF FACSIMILE TRANSMISSION
37 C.F.R. 1.8

The undersigned hereby certifies that this correspondence is being facsimile transmitted to the Patent and Trademark Office (Fax Number 703-872.9306) on November 22, 2004.

Rodney M. Anderson
Registry No. 31,939